Amendment to the Claims:

This listing of claims replaces all prior versions, and listings, of claims in the application:

1. (Previously presented) A system, comprising:

a data source, having a plurality of different lines;

a plurality of programmable delay elements, each coupled to one of said plurality of lines, to control a delay in said one of said lines to produce delayed values;

a register, storing values for said programmable delay elements which respectively control an amount of delay caused by said delay elements;

an arbitration logic, coupled to said plurality of delayed values, and operating to determine relative timing of said plurality of lines, wherein said arbitration logic includes a first element which produces a set of first values for said register, and a second element which determines relative arrival of signals based on said first values, and wherein said arbitration logic dithers between different sets of values, and determines which of said plurality of values produces a best desired result, and stores said best result in said register, wherein said plurality of programmable delay elements thereafter are programmed with values in said register.

2. (Original) A system as in claim 1, wherein there are one of said programmable delay elements for each of said plurality of lines.

3. (Original) A system as in claim 1, wherein said register stores a plurality of values, each of said plurality of values controlling one of said programmable delay elements.

4. (Original) A system as in claim 3, further comprising a non volatile memory, storing said plurality of values.

5-7. (Cancelled)

8. (Original) A system as in claim 1, further comprising a graphics device, and wherein said signals are from said graphics device.

9. (Original) A system as in claim 1, further comprising a non-volatile memory, storing values for said delay elements, and loading said values into said register at a specified time.

10. (Cancelled)

11. (Previously presented) A system as in claim 1, wherein said best result is one where the plurality of delayed signals are received at substantially the same time.

12. (Previously presented) A system as in claim 1, wherein said best result is one where the plurality of delayed signals are received at a time skew that allows certain logic elements to operate correctly.

13-21. (Cancelled)

22. (Currently amended) A method comprising:

receiving a plurality of signals from an external device, each of said plurality of signals related to each other; and

programmably delaying some of said signals relative to others of said signals according to prestored values; and

determining if a system [[event]] crash has occurred, and storing new delay values in a non volatile memory responsive to said system crash occurring.

23. (Original) A method as in claim 22, wherein plurality of signals are signals from a bus.

24. (Original) A method as in claim 23, wherein said plurality of signals are signals from a graphics bus.

25. (Original) A method as in claim 22, wherein said delaying comprises storing delay values in a non volatile memory; and

using said values in said non volatile memory to adjust a value of a programmable delay element.

26. (Cancelled)

27. (Previously presented) A method as in claim 25 wherein said system event is a change in system hardware configuration.

28. (Previously presented) A method as in claim 25, wherein said system event is a system crash.

29. (Previously presented) A method as in claim 25, wherein said reobtaining comprises dithering sets of values in a register that stores values for said programmable delay, determining results, and accepting a set of values which have produced a specified delay.

30. (Previously presented) A method as in claim 29, further comprising storing said values in said non-volatile memory.

31. (Previously presented) A method as in claim 29, wherein said specified delay is a result where there is a minimal delay between arrival of all signals.

32. (Previously presented) A method as in claim 29, wherein said specified delay is a result where there is a specified delay between arrival of all signals which allows for clock skew in at least one specified logic element.

33. (Previously presented) A method of equalizing time delays of signals, comprising:

providing a plurality of signals which are produced in times that are synchronized with one another;

providing sets of values which represent different sets of delay values for said plurality of signals using one of said sets to delay each of said plurality of signals by a respective amount, wherein each of said respective amounts is different then each other respective amount for a different one of said signals based on said delay values;

testing said signals, to determine relative amounts of delays in said signals, to produce said delay amount;

repeating and using said testing to find a best one of said sets; and

using said delay amounts from said best one of said sets to delay said signals.

34. (Previously presented) A method as in claim 33, wherein said plurality of signals are signals from a graphics processing device.

35. (Previously presented) A method as in claim 33, wherein said delaying comprises delaying each of the signals by respective amounts which causes them to arrive at a specified location at substantially similar times.

36. (Previously presented) A method as in claim 35, wherein said delaying comprises delaying said signals by specified amounts which causes them to arrive at said location at specified times which are skewed relative to one another, wherein said skew is related to a clock margin of a system.

37. (Previously presented) A method of setting delays in a system, comprising:

storing values indicative of time delays in a register, said time delays representing delays to be applied to signals to obtain a specified result;

detecting a system crash which indicates that said time delays should be changed;

when said system crash is not detected, using said values in said register to cause signal delays, by applying said values to respective programmable delay elements; and

only when said system crash is detected, using a logic element to determine new delay values and applying said new delay values to said programmable delay elements to cause signal delays based on said new delay values.

38. (Previously presented) A method as in claim 37, wherein said using a logic element comprises applying a plurality of delay values to a plurality of respective programmable delay elements to thereby delay a plurality of lines.

39. (Previously presented) A method as in claim 37, further comprising storing said new delay values in a non-volatile memory.

40. (Cancel)

41. (Currently amended) A method as in claim 37, wherein said system [[event]] crash is an operating system crash.

42-45. (Cancelled)

46. (Previously presented) A system as in claim 1, further comprising a system event detector which produces a system event notification responsive to a specified event in the system; and

wherein said arbitration logic operates to determine said values only responsive to said system event detector.

47. (Previously presented) A system as in claim 46, wherein said system event is a change in hardware.

Kindly add the following new claims

48. (New) An electronic device, comprising:

a first device producing a plurality of first outputs;

a plurality of programmable delay elements, each of said plurality of programmable delay elements connected at one end to one of said plurality of outputs and each producing a second output which is delayed relative to said first outputs;

a levelization register, storing a plurality of values, said values each individually controlling one of said delay

elements to control an amount of delay caused by said delay element to one of said plurality of outputs; and

arbitration logic, connected to each of said second outputs, and determining a relative delay among said second outputs, and wherein said arbitration logic is responsive to a system event flag, which indicates a specified event in the system.

49. (New) A device as in claim 48, wherein said specified event is a hardware change.

50. (New) A device as in claim 48, wherein said specified event is a system crash.

51. (New) A system as in claim 48, wherein said arbitration logic is responsive to said flag to produce a first multiple sets of values for said levelization register, command said first device to produce said signals for each of said sets of values, and determine a relative delay among said signals based on said first sets of values and stores a best set of values as levelization values.

52. (New) A system as in claim 48, further comprising a non volatile memory which stores levelization values, said

arbitration logic storing said levelization values in said non volatile memory.

53. (New) A system as in claim 48, further comprising, at initial system start up, downloading values from said non volatile memory to said levelization register.

54. (New) A system as in claim 48, wherein said programmable delay elements are phase locked loops.